REFERENCES:

[1] A. Myers. JFlow: Practical mostly-static information flow control. In *Symposium on Principles of Programming Languages*, pages 228–241, 1999.

[2] A. Beugnard, J.-M. J´ez´equel, N. Plouzeau, and D. Watkins. Making components contract aware. In *ieee-software*, pages 38–45, june 1999.

[3] Chris Hote. Run-time error detection through semantic analysis: A breakthrough solution to todays software testing inadequacies. Technical report, PolySpace Technologies, September 2004.

[4] D. M. Tullsen, S. Eggers, and H. M. Levy. Simultaneous multithreading: Maximizing on-chip parallelism. In Proceedings of the 22nd Annual International Symposium on Computer Architecture, pages 392{403, June 1995.

[5] D. M. Tullsen, S. J. Eggers, J. S. Emer, H. M. Levy, J. L. Lo, and R. L. Stamm. Exploiting choice: Instruction fetch and issue on an implementable simultaneous multithreading processor. In Proceedings of the 23rd Annual International Symposium on Computer Architecture, pages 191{202, May 1996.

[6] Doron A. Peled. *Software Reliability Methods*. Springer, 2001.

[7] Dr. Linda Rosenberg, Ted Hammer, and Jack Shaw. Software metrics and reliability. Technical report, NASA Software Assurance Technology Center, November, 1998.

[8] Francesca Saglietti. Licensing reliable embedded software for safety- critical applications. Technical report, Department of Software Engi-neering, University of Erlangen-Nrnberg.

[9] F. Walkerden and R. Je®rey. An Empirical Study of Analogy-Based Software E®ort Estimation. *Empirical Software Eng.*, 4(2), pp.135{158, 1999.

[10] G. Ammons, R. Bodik, and J. Larus. Mining specifications.

[11] Giuseppe Lami. QuARS: A tool for analyzing requirements. Techni- cal report, Carnegie Mellon Software Engineering Institute, September 2005.

[12] Ivy Hooks. Writing good requirements. In *Proceedings of the Third International Symposium of the NCOSE*, page Vol. 2, 1993.

[13] J. Tubella and A. Gonzalez. Control speculation in multithreaded processors through dynamic loop detection. In Proceedings of the 4th International Conference on High-Performance Computer Architecture, pages 14{23, January 1998.

[14] J. G. Ste®an and T. C. Mowry. The potential for using thread-level data speculation to facilitate automatic parallelization. In Proceedings of the 4th International Conference on High-Performance Computer Architecture, pages 2{13, January 1998.

[15] J. Xu B. Randell and A. Romanovsky. A generic implementation ap-proach to concurrent fault-tolerant software. Technical report.

[16] J. Aldrich. *Using Types to Enforce Architectural Structure*. PhD thesis, University of Washington, August 2003.

[17] John D. Musa. *Software Reliability Engineering: More Reliable Software Faster and Cheaper*. Autorhouse, second edition, 2004.

[18] J. Hunter and W. Crawford. *Java Servlet Programming*. O'Reilly, second edition, April 2001.

[19] K.DistriNet Research Group. DiPS home page. http://www.cs.kuleuven.ac.be/cwis/research/- distrinet/projects/DIPS/.

[20] M. Acharya, T. Sharma, J. Xu, and T. Xie. Effective generation of interface robustness properties for static analysis. In *Proc. ASE 2006*, pages 293–296.

[21] M. Acharya and T. Xie. Static detection of API errorhandling bugs via mining source code. Technical Report TR-2007-35, North Carolina State University Department of Computer Science, Raleigh, NC, October 2007.

[22] M. Acharya, T. Xie, J. Pei, and J. Xu. Mining API patterns as partial orders from source code: From usage scenarios to specifications. In *Proc. ESEC/FSE 2007*, pages 25–34.

[23] M. Acharya, T. Xie, and J. Xu. Mining interface specifications for generating checkable robustness properties. In *Proc. ISSRE 2006*, pages 311–320.

[24] M. Xie. *Software Reliability Modeling*. World Scienti¯c, Singapore, 1991.

[26] Musa, Iannino, and Okumoto. *Software Reliability Engineering: Measurement, prediction, application*. Mc Graw Hill, 1987.

[27] M. Shaw and D. Garlan. *Software Architecture – Perspectives on an emerging discipline*. Prentice-Hall, 1996.

[28] N. Janssens, S. Michiels, T. Mahieu, and P. Verbaeten. Towards Hot-Swappable System Software: The DiPS/CuPS Component Framework. In *Proceedings - The Seventh International Workshop on Component Oriented Programming*, 2002.

[29] P. Clements. A survey of architecture description languages. In *Proceedings of the 8th International Workshop on Software Specification and Design*, March 1996.

[30] Sun. MAJC architecture tutorial. Technical report, Sun Microsystems Inc., 1999.

[31] S. Yamada, et al. A Software Reliability Growth Model with two Types of Errors. *R.A.I.R.O.*, vol. 19, pp.87{104, 1985.

[32] S. Yamada, H. Ohtera, and H. Narihisa. Software Reliability Growth Models with Testing E®ort. *IEEE Trans. Reliability*, R-35, pp.19-23, April, 1986.

[33] S. Yamada, and S. Osaki. Discrete Models for Software Reliability Evaluation. *In Reliability and Quality control*, Ed. A.P. Basu, Elsevier, London, pp.401{412,

1986.

[34] S. Michiels, L. Desmet, N. Janssens, T. Mahieu, and P. Verbaeten. Self-adapting concurrency: The DMonA architecture. In D. Garlan, J. Kramer, and A. Wolf, editors, *Proceedings of the First Workshop on Self-Healing System (WOSS'02)*, pages 43–48, Charleston, SC, USA, 2002. ACM SIGSOFT, ACM press.

[35] S. Michiels, F. Matthijs, D.Walravens, and P. Verbaeten. Position summary: DiPS: A Unifying Approach for Developing System Software. In A. D. Williams, editor, *Proceedings - The Eigth Workshop on Hot Topics in Operating Systems (HoTOS-VIII*, page 175. University of Karlsruhe, IEEE Computer Society Press, 2001.

[36] T. Tsai and N. Singh. Libsafe 2.0: Detection of format string vulnerability

exploits. White paper, Avaya Labs, February 2001.

[37] T. N. Vijaykumar and G. S. Sohi. Task selection for a multiscalar processor. In Proceedings of the 31st Annual International Symposium on Microarchitecture, pages 81{92, November 1998.

[38] Dr. S. M. K. Quadri et. And al."Improving Software Reliability using Software Engineering Approach- A Review" International Journal of Computer Applications (0975 – 8887) Volume 10– No.5, November 2010.

[39] Dr. S. M. K. Quadri et. And al. "software Reliability Growth Modeling with new modified Weibull testing-effort and optimal release policy" International Journal of Computer Applications (0975 – 8887) Volume 6– No.12, September 2010.